

爱创课堂前端培训

CSS

第一天

班级： 爱创课堂 7 期

日期： 2017 年 4 月 14 日

目录

HTML.....	错误！未定义书签。
目录.....	1

复习

1、列表

无序列表：ul>li。

有序列表：ol>li。

自定义列表：dl>(dt+dd)。

2、div 和 span 是两个布局的标签。

3、表格：table>tr>td

三个划分区域的标签:caption(表格标题)，thead(表头)，tbody（表格主体）

```

1      <table>
2          <caption></caption>
3          <thead>
4              <tr>
5                  <th></th>
6                  <th></th>
7                  <th></th>
8              </tr>
9          </thead>
10         <tbody>
11             <tr>
12                 <td></td>
13                 <td></td>
14                 <td></td>
15             </tr>
16         </tbody>
17     </table>

```

4、表单元素

```

1  单行文本框: <input type="text" />
2  密码框: <input type="password" />
3  单选框: <input type="radio" name="sex" />
4  复选框: <input type="checkbox" name="suibian" />
5  文本域: <textarea cols="30" rows="10">默认文本</textarea>
6  下拉列表:
7      <select name="" id="">
8          <option value=""></option>
9      </select>
10  普通按钮: <input type="button" value="我是一个按钮" />
11  提交按钮: <input type="submit" />
12  重置按钮: <input type="reset" />

```

label 标签:

```

1  <input type="checkbox" checked="checked" id="dianji" />
2  <label for="dianji">同意<a href="">用户须知</a>和<a href="">服务条款</a></label>

```

二、html 杂项

1、注释

注释: 内容不会显示在网页中。

快捷键: ctrl+/

作用: 标注某内容的开始或结束。或者是不想给别人看的代码。

```
1  <!-- 表格标题 -->
```

2、字符实体

一些符号有特殊作用, 直接写在代码中不会显示你想要的效果。

也叫转义字符: 可以直接帮我们输出特殊符号的文本。

常用的一些转义字符:

```

1  空格: &nbsp;
2  大于号: &gt;
3  小于号: &lt;
4  版权信息: &copy;
5  花符号: &amp;

```

实际工作中遇到哪一个可以百度，或者直接查 w3school。

参考手册
HTML 标签列表
HTML 属性
HTML 事件
HTML 合法 DTD
HTML 颜色名
HTML 字符集
HTML ASCII
HTML ISO-8859-1
HTML 符号
HTML URL 编码
HTML 语言代码
HTTP 消息

3、废弃标签

font, **b**, **i**, **u**, **em**, del, strong, br, hr

```
1      <font>这是一个 font 标签</font>
2      <i>我是一个 i 标签</i>
3      <b>我是一个 b 标签</b>
4      <u>我是一个 u 标签</u>
5      <em>我是 em 标签，有强调的语义</em>
6      <u>我是 u 标签</u>
7      <del>我是 del 标签</del>
8      <strong>我是一个 strong 标签，有强调的语义</strong>
9      <hr size="10" color="red" width="500px" />
```

三、css 基础

1、css 概念

css (cascading style sheet) 层叠式样式表。

概念包括：层叠式，样式。

2、css 作用

前端 3 层：

```
1  html      结构层。
2  css       样式层
3  Javascript 行为层
```

css 涉及 2 个部分：选择器（给某个元素或标签设置样式），样式属性（想给选择器添加什么属性就设置什么样式）

```
1      div{
2          width:300px;
3          height:300px;
4          background-color:pink;
5          font-size:20px;
6          font-family:"Microsoft Yahei";
7          color:black;
8      }
```

css 的作用：

- 1) 盒子实体化，进行页面布局。
- 2) 设置文字样式。

3、css 一些样式

```

1      div{
2          /*实体化盒子*/
3          width:300px;
4          height:300px;
5          background-color:pink;
6          border:1px solid black;
7          font-size:20px;
8          font-family:"Microsoft Yahei";
9          color:black;
10     }
```

①实体化盒子:

width:宽度, 单位像素 (px) ;

height: 高度, 单位像素 (px) ;

background-color:背景颜色。

色值表示法:

```

1 十六进制表示法:    #ff0000;
2  rgb 表示法:        rgb(255,0,0);
3  单词表示法:        red;
```

十进制: 0-9

十六进制: 0-9,

a: 10

b: 11

c: 12

d: 13

e: 14

f: 15

```
1  ff= 15*16 + 15 = 255
```

background 是复合属性。

单一属性, 单词间用横线连接。

border:1px solid #ffff00;

线的宽度:1px

线的类型: **solid** 实线, **dashed** (虚线), **dotted**(点线)

线的颜色: #ffff00;

②文字样式:

color:设置文字的颜色。用色值表示法。

```
1  color:#ff0000;    //红色
```

font-size:设置字号。单位是像素 px。

```
1  font-size:20px;
```

font, 是复合属性, 其中的单一属性用短横连接。

font-family: 设置字体。

1 **font-family:** "微软雅黑";

字体: 中文字体, 英文字体。

英文字体: 黑体, **Arial**。

中文字体: 微软雅黑, 宋体。

注意: 1) 我们设置字体时, 因为不同的用户, 字体不一样。我们尽量用计算机自带的字体。

书写时, 先英文字体, 在写中文字体。以防万一, 尽量写一个备用字体, 宋体。

2) 在书写时可以写中文, 但是想提高加载速度, 需要中文换成英文。每个字体用逗号隔开。

1 **font-family:** "Arial", "Microsoft Yahei", "SimSun";

4、css 书写位置

①**行内式**: 标签有一个 **style** 属性, 可以把 **css** 样式写在 **style** 属性里。包括 **css** 全部属性。

但是我们行内式的样式是后台程序员添加。

1 **<p style="color:red;font-size:20px;">**我是一个段落**</p>**

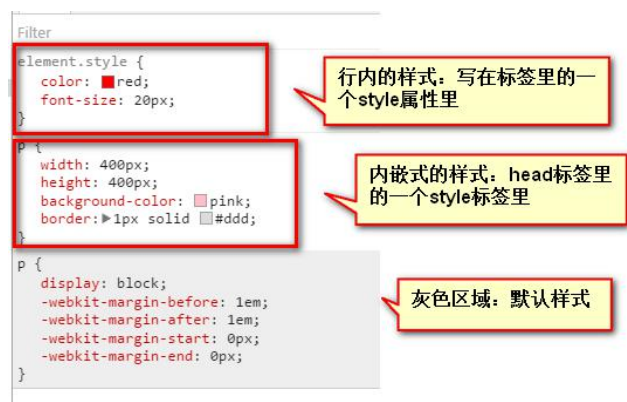
②**内嵌式** (嵌入式): 写在 **head** 标签内部, 一个 **style** 标签里。

```
1 <head>
2   <style type="text/css">
3     p{
4       width:400px;
5       height:400px;
6       background-color:pink;
7       border:1px solid #ddd;
8     }
9     h3{
10      font-size:50px;
11      color:green;
12    }
13  </style>
14 </head>
```

style 标签有一个 **type** 属性。属性值 "**text/css**" 表示 **style** 标签内部书写的是纯文本的 **css** 样式。

type 属性可以不写, 但是要写就必须写对。

③**外链式**。实战课会说。



5、css 杂项

html 中键值对的写法: **k="v"**。

css 中键值对的写法: **k:v**;

css 中分号的重要性：每条样式中间用**分号**分割；每个样式要逐条书写。有什么属性就写样式。

```
1      width:400px;
2      height:400px;
3      background-color:pink;
4      border:1px solid #ddd;
```

css 中，对空格，缩进，换行不敏感。

要求大家，css 代码逐条书写，为了提高代码可读性。

四、选择器

1、基础选择器

选择器：设置 html 元素或者 html 标签的样式。

1) 标签选择器

标签名是什么就写什么。前面不要任何符号。

标签选择器特点：能够选中页面上所有同种标签，不管嵌套多少层都可以被选中。（不太常用）

```
1      <div>
2          <div>
3              <div>
4                  <div>
5                      <p>我是好深深段落</p>
6                  </div>
7              </div>
8          </div>
9      </div>
```

用途：清空我们默认样式，或者是重置默认样式。

```
1      ul{
2          /*去掉小圆点*/
3          list-style:none;
4      }
5      body{
6          background-color:#eee;
7      }
8
```

2) id 选择器

通过标签的 id 属性，进行设置。

书写方法：**#id 名**。#和 id 名之间不要加空格。

```
1      #pa{
2          font-size:30px;
3      }
```

注意：①任何一个标签的 id 名是唯一。即使是不同的标签也不能用相同的 id 名。

②id 名的命名规则：以**字母开头可以有数字、下划线、横线符号**。

层叠性体现 1：同一元素既有标签设置，同时又有 id 设置，这些属性都会体现这个元素上。

```
p{
    color:red;
}
```

```
#pa{
    font-size:30px;
}
<p id="pa">这是一些段落</p>
```

这是一些段落

标签选择器能够选中页面上所有的同种标签。

id 选择器又只能选中一个元素也不好。

3) 类选择器

类选择器：通过元素上面的 **class** 属性的值可以选中**相同类名**的所有标签元素。这是我们实际工作中常用的选择器。

写法：**.类名**

类名命名规则：和 id 命名相同，以字母开头，可以下划线，横线，数字。严格区分大小写。

```
1 <p class="warning">这是一些段落 warning</p>
```

特殊应用：原子类。通常页面会设置一些原子类（公共类），原子类的属性尽可能少。

```
1 .fs20{
2     font-size:20px;
3 }
4 .fs22{
5     font-size:22px;
6 }
7 .red{
8     color:red;
9 }
10 .green{
11     color:green;
12 }
```

一个元素可以有多个类名，类名之间用**空格**隔开。

```
1 <p class="fs20 green">这是一些段落 green</p>
```

类上样式，id 上行为。

4) 通配符

通配符：*，能够选中页面上所有的元素，包括 body。

```
1 *{
2     padding:0;
3     margin:0;
4     font-size:30px;
5     color:#333;
6 }
```

作用：清除默认样式。

但是实际工作不用，因为通配符的效率不高。

2、高级选择器

高级选择器就是基础选择器上进行延伸。

1) 后代选择器

通过有嵌套关系的类选择器或者标签选择器，确认一个大范围，在大范围内进行选择，可以再选择。

写法：用**空格**隔开每一层。

```
1      div p{
2          color:yellow;
3      }
```

在 **style** 标签里设置一段 **css** 样式，

```
1 .box p{
2     color:orange;
3 }
```

下面这段代码可以被选中。后代选择器不但能够选够选中**儿子**还能选中**孙子**，**曾孙子**.....

```
1      <div class="box">
2          <div>
3              <p>我是超微有点靠里的 p</p>
4              <p>我是超微有点靠里的 p</p>
5              <p>我是超微有点靠里的 p</p>
6          </div>
7      </div>
```

后代选择器可以简写，也可以补全。无论如何必须相对应的嵌套关系。

```
1      div .box p{      //补全写法
2          color:red;
3      }
4      div p{
5          color:red;
6      }
```

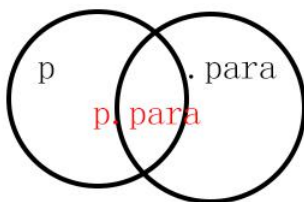
```
1      <div>
2          <div class="box">
3              <p>文字</p>
4              <p>文字</p>
5              <p>文字</p>
6          </div>
7      </div>
```

2、交集选择器

交集选择器：**同时**满足几个条件。

写法：几个选择器的**连写**。不要有任何空格。

小案例：想设置选择器既有 **.para** 类又是 **p** 标签。 **p.para**



```
1      p.para{
2          color:lightblue;
3      }
4      <div>
```



```

5      <p>我是第一层的 p</p>
6      <div class="para">
7          <p>我是第二层的几个 p</p>
8          <p class="para">我是第二层的几个 p</p>
9          <p>我是第二层的几个 p</p>
10     </div>
11 </div>

```

只有涂有颜色的代码会变为浅蓝色。因为，它既是 `p` 标签，本身又有 `.para` 类。

小例子 2:

```

1      .para.fs20.orange{
2          color:purple;
3          font-size:50px;
4      }
5      <div>
6          <p class="fs20 orange">这是一些文字</p>
7          <p>这是一些文字</p>
8          <p>这是一些文字</p>
9          <p class="para fs20 orange">这是一些文字</p>
10         <p>这是一些文字</p>
11         <p class="para fs20">这是一些文字</p>
12         <p>这是一些文字</p>
13         <p class="orange">这是一些文字</p>
14         <h3 class="para fs20 orange">这也是一段文字</h3>
15     </div>

```

满足条件的代码：必须有 3 个条件，缺一不可。

- ①必须有 `.para` 类
- ②必须有 `.fs20` 类
- ③必须有 `orange` 类

```

1      .para.fs20{color:yellow;}    //这种写法是正确的
2      #para.fs20{color:purple;}    //这种写法是错误的，因为 id 唯一。

```

和后代选择器进行区分：

```

1      .para.fs20{color:yellow;}    //这是交集选择器，表示既有 .para 类又有 .fs20 类。
2      .para .fs20{color:green;}    //这是后代选择器，表示 .para 后代里面的 .fs20 类

```

3) 并集选择器

同时选中多个元素。

表示方法：用逗号分隔每个元素，表示“或”的意思。

```

1      p,h3,li{
2          color:purple;
3      }

```

表示 `p,h3,li` 标签字体颜色为紫色。

等价写法：

```

1      p{
2          color:orange;
3      }
4      h3{
5          color:orange;
6      }
7      li{
8          color:orange;
9      }

```

```
9      }
```

类名也可以用并集选择器：表示：有 `.para` 类或者有 `.fs20` 类或者有 `.orange`。

```
1      .para,.fs20,.orange{  
2          color:pink;  
3          font-size:30px;  
4      }
```

等价写法：

```
1 .para{color:pink;font-size:30px;}  
2 .fs20{color:pink;font-size:30px;}  
3 .orange{color:pink;font-size:30px;}
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

1
1
1
1
1
1